## CLAIMS

1. An apparatus comprising:

at least one processor;

a memory coupled to the at least one processor;

a cluster engine residing in the memory and executed by the at least one processor, the cluster engine providing a mechanism for communicating ordered messages to and from a plurality of nodes in a computer cluster, wherein the apparatus comprises one node in the computer cluster;

a protocol residing in the memory that specifies at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received without using any timer; and

a job residing in the memory and executed by the at least one processor, the job processing the protocol, wherein the job functions according to receiver logic that uses the at least one ACK round to determine without using any timer whether the at least one data message has been received.

2. The apparatus of claim 1 wherein the protocol comprises a plurality of phases that are each followed by an acknowledge (ACK) round.

3. The apparatus of claim 2 wherein each phase of the protocol is defined so that no node can both send a data message and receive a data message during any phase of the protocol.

4. The apparatus of claim 1 wherein the receiver logic comprises post-ACK logic that determines whether the at least one data message has been received after the job processes a selected one of the at least one ACK rounds.

5. The apparatus of claim 1 wherein the receiver logic comprises pre-ACK logic that determines whether the at least one data message has been received before the job processes a selected one of the at least one ACK rounds.

6. A networked computer system comprising:

a cluster of computer systems that each includes:

a network interface that couples each computer system via a network to other computer systems in the cluster;

a memory;

a cluster engine residing in the memory that provides a mechanism for communicating ordered messages to and from the computer systems in the cluster;

a protocol residing in the memory that specifies at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received; and

a job residing in the memory and processing the protocol, wherein the job functions according to receiver logic that uses the at least one ACK round to determine without using any timer whether the at least one data message has been received.

7. The networked computer system of claim 6 wherein the protocol comprises a plurality of phases that are each followed by an acknowledge (ACK) round.

8. The networked computer system of claim 7 wherein each phase of the protocol is defined so that no node can both send a data message and receive a data message during any phase of the protocol.

9. The networked computer system of claim 6 wherein the receiver logic comprises post-ACK logic that determines whether the at least one data message has been received after the job processes a selected one of the at least one ACK rounds.

10. The networked computer system of claim 6 wherein the receiver logic comprises pre-ACK logic that determines whether the at least one data message has been received before the job processes a selected one of the at least one ACK rounds.

11. A computer-implemented method for processing a protocol using a plurality of jobs that form a group in a clustered computing environment, the method comprising the steps of:

providing a cluster engine for each member of the group that communicates with the other cluster engines in the group;

defining the protocol so that non-receipt of a message by the cluster engine can be determined without using any timer, the protocol including at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received; and

a job processing the protocol and functioning according to receiver logic that uses the at least one ACK round to determine without using any timer whether the at least one data message has been received.

12. The method of claim 11 wherein the step of defining the protocol further comprises the step of defining a plurality of phases that are each followed by an ACK round.

13. The method of claim 12 wherein the step of defining the protocol further comprises the step of defining each phase of the protocol so that no node can both send a data message and receive a data message during any phase of the protocol.

14. The method of claim 11 further comprising the step of the job using post-ACK logic that determines whether the at least one data message has been received after the job processes a selected one of the at least one ACK rounds.

15. The method of claim 11 further comprising the step of the job using pre-ACK logic that determines whether the at least one data message has been received before the job processes a selected one of the at least one ACK rounds.

16. A program product comprising:

(A) a protocol that specifies at least one data message and at least one acknowledge (ACK) round that provides a time benchmark for determining whether or not a message has been received without using any timer;

(B) a job that processes the protocol, wherein the job functions according to receiver logic that uses the at least one ACK round to determine without using any timer whether the at least one data message has been received; and

(C) computer-readable signal bearing media bearing the protocol and the job.

17. The program product of claim 16 wherein the signal bearing media comprises recordable media.

18. The program product of claim 16 wherein the signal bearing media comprises transmission media.

19. The program product of claim 16 wherein the protocol comprises a plurality of phases that are each followed by an acknowledge (ACK) round.

20. The program product of claim 19 wherein each phase of the protocol is defined so that no node can both send a data message and receive a data message during any phase of the protocol.

21. The program product of claim 16 wherein the receiver logic comprises post-ACK logic that determines whether the at least one data message has been received after the job processes a selected one of the at least one ACK rounds.

22. The program product of claim 16 wherein the receiver logic comprises pre-ACK logic that determines whether the at least one data message has been received before the job processes a selected one of the at least one ACK rounds.

*****